Process Scheduling

Process Scheduling

- In multiprogramming environment, multiple processes are maintained in main memory.
- Each process alternates between **using a CPU** and **waiting for some I/O** to be performed or some event to occurs.
- With this, CPU is kept busy by executing one process while other wait.
- The key to multiprogramming is **scheduling of process**

- The first three types mentioned are the strategies used for CPU scheduling.
- What is CPU scheduling?
  - The assignment of CPU to processes allows CPU to accomplish the task/work.
  - The probability of determining when CPU should be assigned and to which processes is called CPU scheduling.
  - Part of CPU concerned with the decision is called scheduler and algorithm is used is called scheduling algorithm.

Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

Note usage of the words **DEVICE, SYSTEM, REQUEST, JOB.**

UTILIZATION	The fraction of time a device is in use. (ratio of in-use time / total observation time)
THROUGHPUT	The number of job completions in a period of time. (jobs / second)
SERVICE TIME	The time required by a device to handle a request. (seconds)
QUEUEING TIME	Time on a queue waiting for service from the device. (seconds)
RESIDENCE TIME	The time spent by a request at a device. RESIDENCE TIME = SERVICE TIME + QUEUEING TIME.
RESPONSE TIME	Time used by a system to respond to a User Job. (seconds)
THINK TIME	The time spent by the user of an interactive system to figure out the next request. (seconds)

The goal is to optimize both the average and the amount of variation. (but beware the ogre predictability.)

Criteria of CPU/processor scheduling

- Different scheduling alog. have different properties and hence we have to choose the algorithm according to the way the processes in the system.
- Care must be taken to choose the right algorithm that satisfy the requirement.
- These requirements are
  - Fairness
  - Response time
  - Waiting
  - Turnaround time etc.

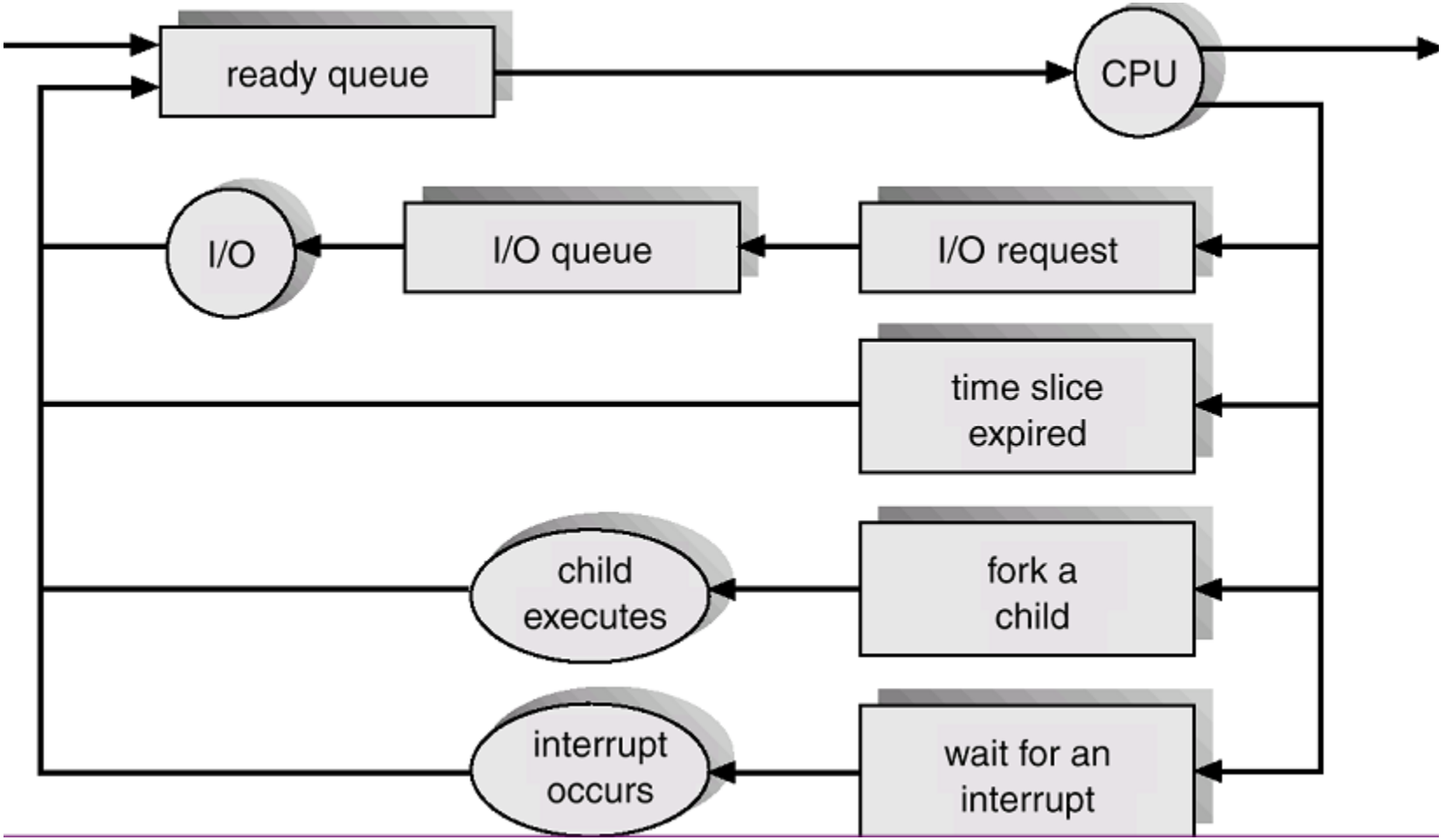
- Based on the system, scheduling algorithm must be selected like batch system, time sharing system, real time system etc.
- Some key points:
  - Fairness :
    - A scheduler makes sure that each process gets its fair share of cpu and no process can suffer indefinite postponement.
  - Policy enforcement:
    - Scheduler must ensure that system's policy is enforced.
    - Means if the local policy is safety then the safety control processes must be able to run whenever they want to.

- Efficiency:
  - Scheduler should keep the system busy all the time.
  - If cpu and i/o devices are kept busy then more work gets done per second.
- Response time:
  - This measure is the time from the submission of process till the first response is produced.
  - Scheduler should minimize the response time for interactive user.
- Turnaround time:
  - The interval from the time of submission of a process to the time of completion of the process.
  - Turnaround time = total waiting time + waiting time in ready queue + total execution time

- Throughput:
  - The measure of the work gets done bye the cpu
  - i.e. number of processes completed in second.
- CPU utilization:
  - CPU should be kept as busy as possible
  - It can have value in the range 0 to 100.

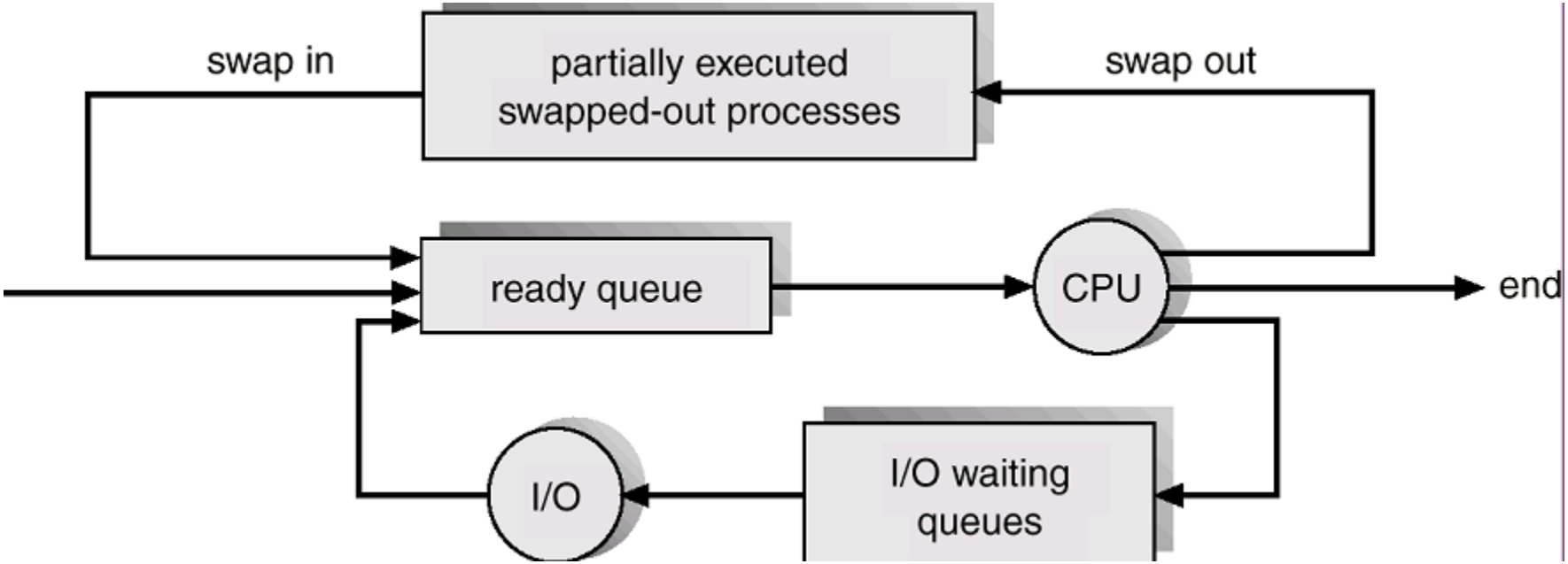
Types of scheduler

- Long term scheduler :
  - Determines which programs are admitted to the system for processing.
  - It controls the degree of multiprogramming.



- Select which processes should be brought into the ready queue.
- At this stage, scheduler must decide that the os can take one or more additional processes.
- It must decide which job to accept and run into CPU.

- Medium term scheduler :



- It is the part of swapping function.
- Swapping in decision is based on the need to manage the degree of multiprogramming.

- Short term scheduler :
  - It can be called as CPU scheduler
  - selects which process should be executed next and allocates CPU

- Short-term scheduler is invoked very frequently (milliseconds) ➡ (must be fast).
- Long-term scheduler is invoked very infrequently (seconds, minutes) ➡ (may be slow).
- The long-term scheduler controls the degree of multiprogramming.
- Processes can be described as either:
  - ✦ I/O-bound process – spends more time doing I/O than computations, many short CPU bursts.
  - ✦ CPU-bound process – spends more time doing computations; few very long CPU bursts.

Sr. no.	Type	Description
1	Long term scheduling	The decision to add to the pool of processes to be executed
2	Medium term scheduling	The decision to add to the number of processes that are partially or fully in main memory.
3	Short term scheduling	The decision as to which available process will be executed by the CPU.
4	I/O scheduling	The decision as to which process's pending I/O request shall be handled by an available I/O devices.

Context switching

- It is an essential feature of the multitasking OS.
- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process.
- Context-switch time is overhead; the system does no useful work while switching.
- Context switching can mean task context switching, thread context switching of process context switching.
- Time dependent on hardware support.

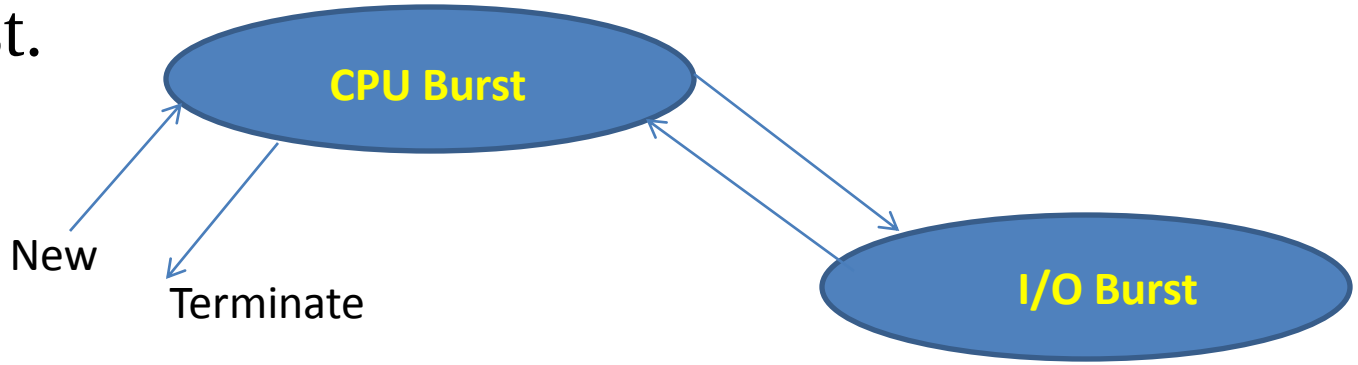
- Context switching steps :
- Step 1:
  - State of the process includes all the register that the process may be using, especially the Program Counter
  - And any other necessary data
- Step 2:
  - All these stored in one data structure called switch frame of PCB.
- Step 3:
  - In order to switch the processes, PCB of first process must be created and saved.
  - Load the next process data like is PCB and maintain the PC information.
- Context switching can be implemented with software as well as with hardware.

- Preemptive scheduling
  - It add the cost to the system when it preempts the process which is about to complete.
  - It also affects the kernel design

- Non preemptive scheduling
  - Short jobs are made to wait by longer jobs but overall treatment of all processes is fair.
  - Response times are more predictable.

Process Scheduling Algorithms

- Process execution consists of a cycle of CPU execution and I/O wait.
- Process alternates between this two states.
- Process execution starts with CPU burst followed by I/O burst and the sequence of alternate CPU and I/O burst goes on.
- Last cpu burst will terminate the process.
- CPU bound programs tend to have a few very long cpu burst and I/O bound programs have many short CPU burst.

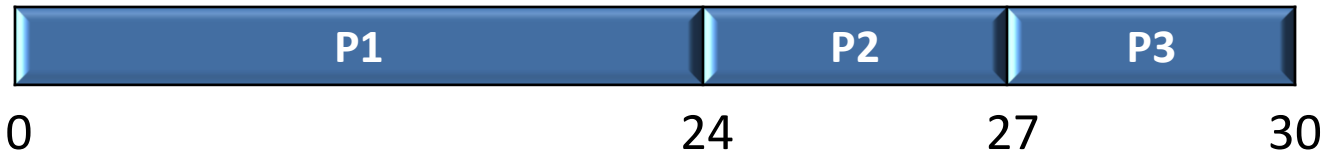


Process Scheduling Algorithms

- FCFS – First Come First Served algorithm
  - When the process enters in a ready queue,
    - its PCB is linked on to the tail of the queue
    - When the cpu is free it is allocated to the process at the head of the queue

PROCESS	BURST TIME
P1	24
P2	3
P3	3

- Consider the set of processes that arrive at time 0.
- Process P1, P2 and P3 arrive in order and served in FCFS manner then the total
- With the use of chart,



Process	Wait time	Turn around time
P1	0	0+24 = 24
P2	24	24 + 3 = 27
P3	27	27 + 3 = 30

Average waiting time = (0 + 24 + 27) / 3 = 17 ms

There are many **drawbacks** in this type of scheduling algorithm

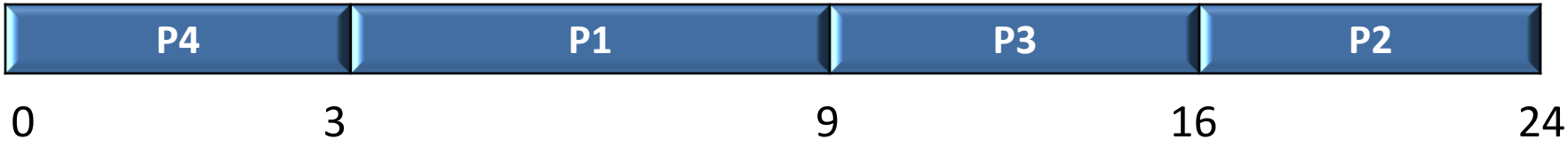
1. Waiting time for individual process is very long.
2. All processes are in queue, so if large process is there in execution then all small processes has to wait for longer time.
3. CPU bound process can complete its execution very fast, but in that time I/O bound processes has to wait
    1. These I/O bound processes will hold the I/O resources, which are in idle condition.
    2. If I/O bound process busy with I/O devices, till that time CPU will remain in idle condition.
4. So, with the use of such kind of algorithm, it lowers CPU utilization and device utilization

Process Scheduling Algorithms

- SJF – Shortest Job First algorithm
  - Preemptive or non preemptive kind of algorithm.
  - In which waiting job with the smallest estimated runtime to completion is run next.

PROCESS	BURST TIME
P1	6
P2	8
P3	7
P4	3

- This is non-preemptive SJF algo.
- Consider the set of processes that arrive at time 0.
- Process P1, P2, P3 and P4 arrive in order and select the process with smallest burst time.
- With the use of chart,



Process	Wait time
P1	3
P2	16
P3	9
P4	0

Average waiting time = (3 + 16 + 9 + 0) / 4 = 7 ms

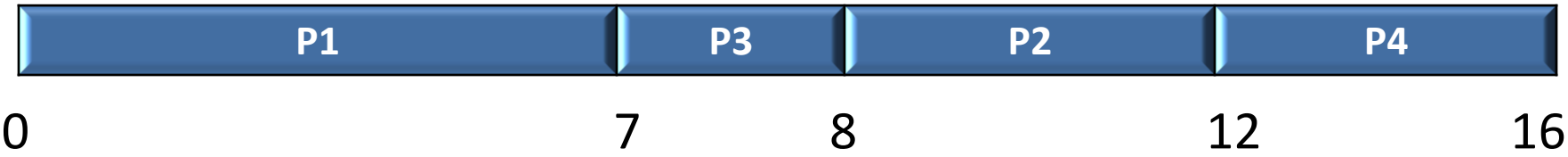
- Some issues with SJF

1. It is appropriate for batch jobs for which run times are knows in advance.
2. Problem with SJF is that it requires precise knowledge of how long a job or process will run and this information is not generally available.

Example #2: Non-Preemptive SJF (varied arrival times)

PROCESS	Arrival Time	BURST TIME
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4

- SJF (non-preemptive, varied arrival times)



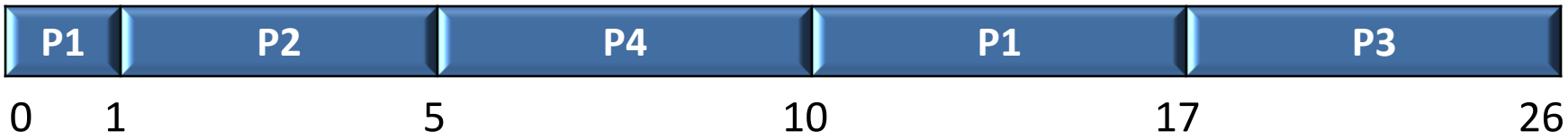
- Average waiting time
  = ((0 – 0) + (8 – 2) + (7 – 4) + (12 – 5))/4
  = (0 + 6 + 3 + 7)/4 = 4
- Average turn-around time:
  = ((7 – 0) + (12 – 2) + (8 - 4) + (16 – 5))/4
  = (7 + 10 + 4 + 11)/4 = 8

Waiting time : sum of time that a process has spent waiting in the ready queue

Process Scheduling Algorithms

- SJF – Shortest Job First algorithm
  - Preemptive algo.
    - Consider the set of processes that arrive at time 0.
    - Process P1, P2, P3 and P4 arrive in order of arrival time.
    - With the use of chart,

PROCESS	BURST TIME	ARRIVAL TIME
P1	8	0
P2	4	1
P3	9	2
P4	5	3



- Process P1 is started at time 0. Process P2 arrive at time 1. The remaining time for processes P1 (7ms) is larger than the time required by process P2 (4 ms) so P1is preempted and P2 is scheduled.

Process	Wait time
P1	(10 -1) = 9
P2	(1-1) = 0
P3	(17 - 2) = 15
P4	(5 - 3) = 2

Average waiting time = (9 + 0 + 15 + 2) / 4 = =6.5 ms

- Some issues with SJF

1. Problem with preemptive SJF is that small process will run immediately, while longer processes have longer waiting time.

Process Scheduling Algorithms

- Priority Scheduling Algorithm
  - Priority is associated with each process and CPU is allocated to the process with highest priority.
  - Equal priority processes are scheduled if FCFS manner.
  - Low number means higher priority.

PROCESS	BURST TIME	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

- Consider the set of processes that arrive at time 0.
- Process P1, P2, P3, P4 and P5 arrive in order and served based on priority manner.
- With the use of chart,

Process	Wait time
P1	6
P2	0
P3	16
P4	18
P5	1

Average waiting time = (6 + 0 + 16 + 18 + 1) / 5 = 8.5 ms

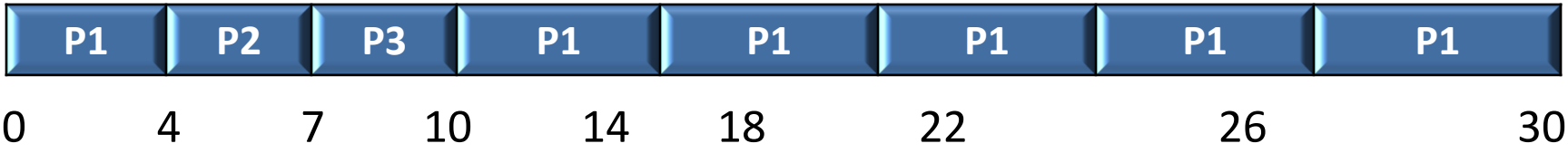
- Some issues with Priority based scheduling

1. Main problem with this algo is infinite blocking of starvation. Means process is ready to run but wont get CPU for longer period of time.
2. This problem can be solved with the **aging technique.**
3. **Aging** is the technique of gradually increasing the priority of processes that wait in system for long time.

- Round Robin (RR) algorithm
  - Used specially for time sharing system.
  - Small unit called time slice is defined called **time quantum**.
  - The ready queue is treated as a **circular queue**.
  - The CPU scheduler goes around the ready queue allocating the CPU to each process for time interval up to 1 quantum

PROCESS	BURST TIME
P1	24
P2	3
P3	3

- Consider the set of processes that arrive at time 0.
- Process P1, P2 and P3 arrive in order
- Consider **time slice of 4 ms**



If there are n processes in the ready queue and time quantum is q, then each process gets of the CPU time in chunks at most q time units.

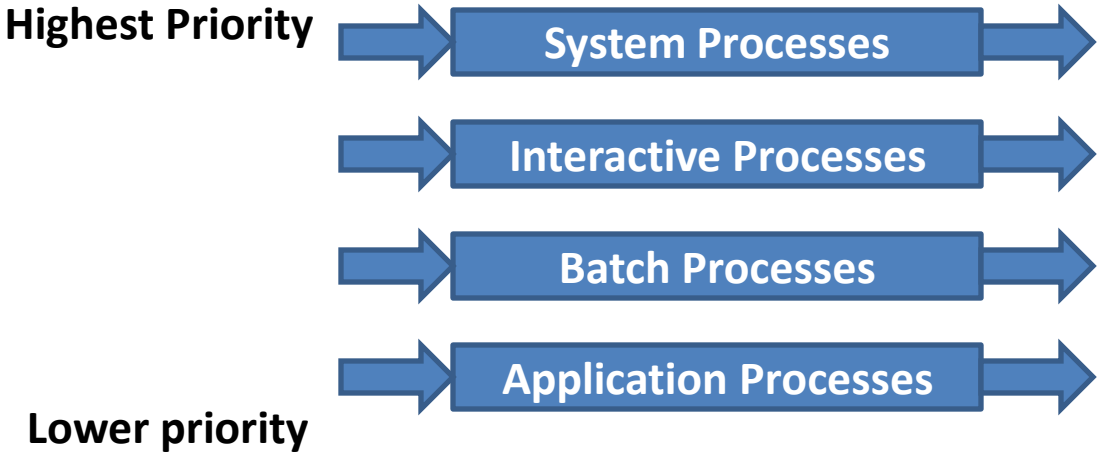
• Multilevel queue scheduling

There are two types of processes :

• Foreground process

• Background process

- These two processes have different response time requirements and different scheduling needs.
- This algorithm partitions the ready queue into several separate queues.
- The processes are permanently assigned to one queue based on some property of the queue.



Multilevel feedback queue scheduling

- It is similar to multilevel queue, but this scheduling allows processes to move between queues.
- If a process uses too much CPU time, it will be moved to a lower priority queue.
- Similarly if process waits too long in a lower priority queue may be moved to higher priority queue
- This form of aging prevents starvation.